

REMARKS

As an initial matter, Applicants respectfully submit that Box 4 and Box 6 of the "Office Action Summary" indicate that claims 1-14 and 18-32 are rejected in the Application. Applicants respectfully submit that claims 1-9 were cancelled by the Amendment filed November 20, 2006, and are not pending in the Application. Therefore, Applicants assume in the response set forth below that the Examiner meant to list claims 10-14 and 18-32 as being rejected.

Claims 10-14 and 18-32 are pending in the Application and all were rejected in the final Office action of August 24, 2007. Claims 10 and 18 have been amended. Claims 10, 18, 26 and 28 are independent claims. Claims 11-14, 19-25, 27 and 29-32 depend either directly or indirectly, from independent claims 10, 18, 26 and 28, respectively.

Applicants respectfully request reconsideration of claims 10-14 and 18-32, in light of the following remarks.

Rejection of Claims

Claims 10-12, 14 and 18-25 were rejected under 35 U.S.C. §102(e) as being anticipated by Spencer et al. (US 2003/0131226, hereinafter "Spencer"). Claims 13, and 26-32 were rejected as being unpatentable over Spencer in view of Yang (US Published Application 2004/0040020). Claims 10-12, 14 and 18-25 were also rejected under 35 U.S.C. §102(b) as being anticipated by Sharon Peleg (PCT WO 0011549). Claims 13, and 26-32 were rejected as being unpatentable over Sharon in view of Yang (US Published Application 2004/0040020).

Office action Applicants respectfully traverse the rejections, for at least the following reasons:

I. Spencer Does Not Anticipate Claims 10-12, 14 and 18-25.

Applicants first address the rejection of claim 10, which has been amended to clarify the subject matter of the claim. The instant Office action repeats the rejection of claim 10 set forth in the Office action of February 7, 2007, maintaining that Spencer, at

paragraphs [0008]-[0009], teaches "...retrieving an existing version of code and a corresponding updated version of the code...", as recited in Applicants' amended claim 10. Applicants respectfully disagree. According to Spencer, at paragraph [0008]:

There are also systems that automatically inform a user about when updates to software and firmware components are available. One such system is the Microsoft Windows Update system. In this system, a user navigates to a product update page in his or her web browser and selects to check for updates. A program component scans the user's system to see what components are installed, and compares the result of the scan with a list of available updates. Any components that do not exist on the user's system or for which updates are available are presented to the user in a list of downloadable options. The user selects which components he or she would like to install, and the components are downloaded and installed on the system. Even though a system like this facilitates installation of program components, the system is limited to a specific vendor and does not consider if the user tries to perform a specific action or not. It merely presents a list of available update options to the user.

Spencer also states, at paragraph [0009]:

In general, in one aspect, this invention provides methods, apparatus, and systems, including computer program products, implementing and using techniques for updating a configuration of a device. An indication of an intended action to be performed by the device is obtained. A configuration list of program components required to perform the intended action is prepared based on the intended action and configuration information for the device. For each program component in the configuration list it is determined whether the program component is installed on the device. Each required program component that is not installed on the device is transferred to the device and the program component is installed on the device.

Applicants respectfully maintain that the above portions of Spencer, specifically cited by the Office action as teaching "...retrieving an existing version of code and a

corresponding updated version of the code...", make no mention of retrieving any version of code, let alone retrieving an existing version of code and a corresponding updated version of the code. Spencer is simply silent in this regard.

The Office action also cites paragraphs [0019], [0030], and [0040], as teaching "...retrieving an existing version of code and a corresponding updated version of the code...." We first turn our attention to Spencer at paragraph [0019], which states:

A "goal-driven" configuration is created in which user does not need much knowledge to download and install components that are necessary to perform certain actions on certain types of files, such as playing media files. The configuration manager on the media playback device can keep a current list of installed components and file formats that can be played on the media playback device. If the user selects a media file format that cannot be played, the configuration manager works in conjunction with a remote system and automatically downloads and installs all the necessary components, before downloading the content itself. If the user selects the media file and wishes to perform an action on the media file that is not currently supported by the device, such as transferring the media file from one device to another device, the necessary components for performing this operation will automatically be downloaded and installed on the device prior to performing the action. A faster and less error-prone installation process with little or no user intervention can be achieved compared to when a user must manually find the necessary components, select the components, download the components to his or her playback device, and install the components. The installation process can iteratively discover if additional components are needed. If additional components are required, the process can automatically obtain and install the additional components in the correct order. Software and firmware components can be obtained from any server that is connected to the communications network and that gives a user permission to download and install components on his or her computer or playback device.

Applicants respectfully submit that this portion of Spencer fails to teach or suggest anything with respect to "...retrieving an existing version of code and a corresponding updated version of the code...." No mention is made of retrieving code,

let alone retrieving an existing version and a corresponding updated version of code. Instead, this portion of Spencer simply teaches that a configuration manager on the media playback device can keep a current list of installed components and file formats that can be played on the media playback device, and if a user selects a media file format that cannot be played, the configuration manager works in conjunction with a remote system and automatically downloads and installs all the necessary components, before downloading the content itself. This portion of Spencer does not, however, teach or suggest "...retrieving an existing version of code and a corresponding updated version of the code...", as recited in amended claim 10.

Next, we turn to Spencer at paragraph [0030], which states:

On the local side of the delivery system, a configuration manager (120, 125) is designed to communicate with the application server (140). The configuration manager (120, 125) can be located in a playback device (105, 110), or in a pass-through device (115). It should be noted that the pass-through device (115) generally also can function as a playback device. For reasons of simplicity, references will be made below to playback devices, but all such references should be considered to include pass-through devices having the same or similar capabilities as playback devices. The configuration manager (120, 125) contains the functionality required for receiving a list from the server of program components that are required by the playback device to perform a specific action requested by the user. The configuration manager can also detect currently installed software and firmware components on the playback device, compare the list of program components with the currently installed program components, request missing program components, and install the program components on the playback device after they have been received from a remote system. Optionally the configuration manager can also perform scheduling of program component downloads, for example, to schedule downloads for a particularly convenient time, and perform maintenance on installed software components, such as version upgrades, and so on. The functionality of the configuration manager will be described in further detail below. The configuration manager can be implemented in

software or firmware, so that it can be implemented on devices with or without full operating systems.

Applicants respectfully submit that this portion of Spencer also fails to teach or suggest anything with respect to "...retrieving an existing version of code and a corresponding updated version of the code...." Instead, this portion of Spencer simply teaches that a configuration manager is capable of receiving a list of program components from a server, can detect currently installed software and firmware components on a playback device, can compare the list of program components with the currently installed program components, can request missing program components, and can install the program components on the playback device after they have been received from a remote system. This portion of Spencer also does not, however, teach or suggest "...retrieving an existing version of code and a corresponding updated version of the code...", as recited in Applicants' amended claim 10.

Finally, we address the teachings of paragraph [0040] of Spencer, which states:

If the device is not properly configured, the configuration manager presents a list of the missing program components, or program components whose version is older than the acceptable version, to the user and receives a user selection of program components to be transferred to the device (225). The list of missing program components is based on the configuration list sent out from the remote system and contains the requested and recommended program components for the device and the type of action. When the user has made a selection of program components in the missing program component list, the selected program components are transferred to the configuration manager from the remote system or from the various locations provided in the configuration list (230). The configuration manager installs the transferred program components on the device after the program components have been received according to instructions that are included in the configuration list that was received by the configuration manager in step 210. The instructions include, for example, in what order the program components should be installed. Alternatively, the program components can be temporarily stored on the device and the installation can be initiated by a user at a different time. After

the program components have been installed, the process checks again if the device is configured according to the configuration list (235). If the device is properly configured, the process indicates to the user that the device is configured to perform the action on the selected media file (240) and continues to step 220, where it performs the desired action when appropriate, and otherwise the process returns to step 230 where it continues to transfer the selected program components, as described above.

Applicants respectfully submit that this portion of Spencer also fails to teach or suggest anything with respect to "...retrieving an existing version of code and a corresponding updated version of the code...." Instead, this portion of Spencer simply teaches that if a device is not properly configured, a configuration manager presents a list of missing program components, or program components older than an acceptable version, to a user and receives user selected program components transferred to the device. The list of missing program components is based on a list sent from the remote system and contains the requested and recommended program components for the device and the type of action. This portion of Spencer also does not, however, teach or suggest "...retrieving an existing version of code and a corresponding updated version of the code...", as recited in Applicants' claim 10.

Applicants respectfully submit that neither the portions of Spencer specifically identified in the Office action, discussed above, nor any other portion or figure of Spencer teaches or suggests "...retrieving an existing version of code and a corresponding updated version of the code...", as recited in Applicants' amended claim 10.

In response to Applicants' arguments filed May 18, 2007, the Office action states "...[the applicant] indicates that the specific portion of retrieving an existing code and an updated version of the code has not been taught; however, although the feature is not cited in the same manner as the applicant, the feature is considered inherent to enable comparisons to determine the missing components, see again sects. 0019, 0030 (retrieves previous version, "existing code", and compare with updated components,

“updated version”) and 0040...” (emphasis added)(Office action at page 3) The Office action also states that “...[t]he applicant is correct in the [sic] Spencer teach [sic] preparing a list of missing components; however, in order to perform this function, the previous components (previous version) have to be known and the newly added features (updated version) has [sic] to be known to determine the differences (missing components). Therefore, it is considered that Spencer inherently acquires (retrieves) both versions to determine the missing components.” (emphasis added)(Office action at page 3) Applicants appreciate recognition in the Office action that “...the feature is not cited in the same manner as the applicant...”, however, Applicants respectfully disagree with the assertion that Applicants’ limitation “...retrieving an existing version of code and a corresponding updated version of the code...” is inherent in the teachings of Spencer.

According to MPEP §2112, Sec. IV, page 2100-54,55, “[t]o establish inherency, the extrinsic evidence ‘must make clear that the missing descriptive matter is **necessarily** present in the thing described in the reference, and that it would be so recognized by persons of ordinary skill. Inherency, however, **may not be established by probabilities or possibilities**. The mere fact that a certain thing **may** result from a given set of circumstances **is not sufficient**.” (emphasis added)

Applicants respectfully submit that the Office action has failed to set forth any basis or reasoning for the assertion that “...retrieving an existing version of code and a corresponding updated version of the code...” is necessarily present in the teachings of Spencer. Applicants respectfully submit that for a particular teaching to be inherent, it must always be so. Applicants respectfully submit that there are ways in which Spencer may determine the missing components that are different from “...retrieving an existing version of code and a corresponding updated version of the code....” Indeed, the Examiner’s cited sections of Spencer illustrate at least one method of making such a determination. For example, Spencer clearly states, in paragraph [0008], that “[a] program component scans the user’s system to see what components are installed, and compares the result of the scan with a list of available updates. Any components that do not exist on the user’s system or for which updates are available are presented to the user in a list of downloadable options....” Applicants respectfully submit that such a

scan may be as simple as looking at a list of files that are present on the system, and comparing that list of files to a list of needed files. Such a list of needed files may be provided by a producer of software updates to be downloaded to the user's system. A similar approach may be used for program components that are not files. For example, a list of functions used by a program may be generated by a compiler or linker. Such a list may be compared to a list of functions already available on a device. Applicants respectfully submit that "...retrieving an existing version of code and a corresponding updated version of the code..." need not necessarily be performed in making such a determination, and that it is not true that determining missing components inherently includes "...retrieving an existing version of code and a corresponding updated version of the code...." Therefore, Applicants respectfully submit that "...retrieving an existing version of code and a corresponding updated version of the code..." is not inherent in the teachings of Spencer.

Based at least upon the above, Applicants respectfully submit that Spencer fails to teach or suggest the feature "...retrieving an existing version of code and a corresponding updated version of the code..." contrary to the assertions in the Office action. Further, Applicants respectfully submit that Spencer fails to teach or suggest "...retrieving an existing version of code and a corresponding updated version of the code, the existing and the updated versions of code comprising program components each comprising a plurality of program instructions...", as recited in Applicants' amended claim 10.

In addition, Applicants respectfully submit that Spencer fails to teach or suggest "...generating an associated reference lookup table having entries corresponding to those program components to be modified, each entry associating a symbolic reference in the updated version of code with a memory address in the mobile electronic device...", as recited in amended claim 10. The Office action asserts that "...generating an associated reference lookup table having entries corresponding to those program components to be modified, each entry associating a symbolic reference with a memory address..." is taught by Spencer, at paragraph [0040], and that "...[t]he reference

lookup table is considered represented by the “configuration list” in sect. 0040...” of Spencer. (Office action at page 3) Applicants respectfully disagree.

According to Spencer, at paragraph [0040]:

If the device is not properly configured, the configuration manager presents a list of the missing program components, or program components whose version is older than the acceptable version, to the user and receives a user selection of program components to be transferred to the device (225). The list of missing program components is based on the configuration list sent out from the remote system and contains the requested and recommended program components for the device and the type of action. When the user has made a selection of program components in the missing program component list, the selected program components are transferred to the configuration manager from the remote system or from the various locations provided in the configuration list (230). The configuration manager installs the transferred program components on the device after the program components have been received according to instructions that are included in the configuration list that was received by the configuration manager in step 210. The instructions include, for example, in what order the program components should be installed. Alternatively, the program components can be temporarily stored on the device and the installation can be initiated by a user at a different time. After the program components have been installed, the process checks again if the device is configured according to the configuration list (235). If the device is properly configured, the process indicates to the user that the device is configured to perform the action on the selected media file (240) and continues to step 220, where it performs the desired action when appropriate, and otherwise the process returns to step 230 where it continues to transfer the selected program components, as described above.

Applicants respectfully submit that the portion of Spencer shown above, which was specifically cited in the Office action, fails to teach or suggest, at least, “...generating an associated reference lookup table having entries corresponding to those program components to be modified, each entry associating a symbolic reference

in the updated version of code with a memory address in the mobile electronic device...”, as recited in amended claim 10. Instead, the cited portion of Spencer simply teaches that if a device is not properly configured, a configuration manager presents a list of missing program components, or program components older than an acceptable version, to a user and receives user selected program components transferred to the device. The list of missing program components is based on a list sent from the remote system and contains the requested and recommended program components for the device and the type of action. The Office action states that “...in reference to each entry associating a symbolic reference to a memory address is considered provided by program components located anywhere throughout the program to enable the items to be located for comparison purposes to determine differences...” (Office action at page 4) Applicants respectfully submit that the intended meaning of this statement is unclear, and respectfully request clarification of how “...program components located anywhere throughout the program to enable the items to be located for comparison purposes to determine differences...” teaches applicants’ feature “...generating an associated reference lookup table having entries corresponding to those program components to be modified, each entry associating a symbolic reference in the updated version of code with a memory address in the mobile electronic device...”, as recited in amended claim 10.

The Office action also suggests that the Applicants ‘...see Spencer’s sect 0015 and 0036 (which specifically references the registry database that is checked for a list of installed values, “which may have to be updated”).’ (Office action at page 4) The Office action earlier suggested correspondence between Applicants’ “reference lookup table” and the “configuration list” of Spencer. (Office action at page 3) From the statement above, the Office action now appears to be suggesting correspondence between Applicants’ “reference lookup table” and a “registry database”. Applicants respectfully submit that the Office is inconsistent in identifying the teachings of Spencer that correspond to Applicants’ element “reference lookup table”.

Applicants respectfully submit that Spencer paragraph [0015] states:

Advantageous implementations can include one or more of the following features. The application server can further include means for transferring one or more required program components to the device. The application server can further include means for storing the configuration list. The application server can further include means for obtaining information about the file type. The file type can include one or more of codec type, codec version, digital rights management type, digital rights management version, encoded bit rate and rights associated with the media file. The means for preparing can include means for querying one or more databases based on information about an intended action to be performed on at least one media file by the device, information about the file type for the media file, and information about the device, and means for assembling result from database queries into a configuration list. The application server can further include means for obtaining information about the device. The configuration list can include one or more of program component name, program component version, registry check key, registry check value, file check location, file check version, file download size, file download URL, and file execution parameters.

Applicants respectfully submit that Spencer paragraph [0036] states:

After this information, which all relates to the track, a section of information follows that relates to the software that needs to be installed on the device in order for the device to be able to download the track. There are two required items, Item 1 and Item 2, that are each defined by a number of parameters. The first required item, Item 1, is a Configuration Installer, version 1.01, with the SoftwareNameID 12345 and a size of 1200 kB. The parameter SoftwareDownloadURL indicates that the software component can be obtained from the URL http://server_name/CfgManInstall.exe. The FileCheck section lists information about the required file, such as the file name "CfgMan.dll," the file type "Win32 Driver," and the FileMinVersion "1.0.2.1," which indicates to the configuration manager that version 1.0.2.1 or later of the .dll file needs to be present on the device. If the configuration manager determines that an older version is present, or that the .dll file is missing on the device, an upgrade is needed. The second required item, Item 2, is is a Windows Media DRM, version 2.1.0, with the SoftwareNameID 12346 and a size of 3255 kB.

The parameter SoftwareDownloadURL indicates that the software component can be obtained from the URL `http://server_name/WMDRMInstall.exe`. In addition to the FileCheck section that lists information about the required file, such as the file name "WMDRM.dll," and the file type "Win32 System," there is a RegistryCheck section that contains instructions for checking for certain values that, for example, represent installed program components, in a registry database on the device. In the present example, the ASX file contains an instruction to check if the parameter `HKLM\Software\DRM\WMDRM.\IsInstalled=1` exists in the registry database.

Applicants respectfully submit that the neither of the cited portions of Spencer shown above, nor any other portion or figure of Spencer teaches or suggests, at least, "...generating an associated reference lookup table having entries corresponding to those program components to be modified, each entry associating a symbolic reference in the updated version of code with a memory address in the mobile electronic device...", in accordance with amended claim 10. Spencer is silent with respect to "...associating a symbolic reference in the updated version of code with a memory address in the mobile electronic device...", and therefore fails to teach or suggest generating a table of such information. Therefore, Spencer fails to teach or suggest at least this aspect of Applicants' amended claim 10.

In addition, Applicants respectfully submit that Spencer fails to teach, suggest, or disclose "...communicating the reference lookup table to the mobile electronic device, to enable the mobile electronic device to resolve the symbolic references in the program instruction to a memory address in the mobile electronic device, during program instruction execution...", as recited in Applicants' amended claim 10. Spencer is simply silent in this regard.

Based at least upon the above, Applicants respectfully submit that the Office action has failed to set forth a reasoned showing where Spencer teaches each and every element of Applicants' amended claim 10, as required by M.P.E.P. §2131, that the Office has failed to establish a *prima facie* case of anticipation, and that a rejection of amended claim 10 under 35 U.S.C. §102(e) cannot be maintained.

Therefore, Applicants believe that amended claim 10 is allowable over Spencer, for at least the reasons set forth above. Applicants respectfully submit that claims 11-14 depend from claim 10, and are also allowable, for at least the same reasons. Applicants therefore respectfully request that the rejection of claims 10-14 under 35 U.S.C. §102(e) be reconsidered and withdrawn.

Applicants now address the rejection of claim 18, which has been amended to clarify the subject matter of the claim. The instant Office action rejects claim 18 "...as claim 1, in view of fig. 1 and sect. 0047." (Office action at page 5) Applicants respectfully submit that there is no pending claim 1. The Office further action states "...[t]he electronic device is considered to consist of the processor and the primary and secondary memory; since items like the lookup table and the management unit does not create or enable a new device. Furthermore software stored in memory also does not enable a new device and the processor and memories appear to be individual components that do not interact until claim 20. **The [Applicant] claims the Spencer [reference] does not teach the reference lookup table management unit; however, the feature is considered taught via Spencer's [sic] configuration unit and the lookup table feature has been addressed above.**" (emphasis in original)(Office action at page 5) Applicants respectfully disagree with the statements in the Office action.

Applicants respectfully submit that Spencer describes FIG. 1 as "...[a] schematic view of a system for delivering audio files, software components and firmware components to a playback device in accordance with invention...." (paragraph [0025])

Applicants respectfully submit that nothing in FIG. 1 or the related text, nor any other portion or figure of Spencer, teaches or suggests, at least, "...a reference lookup table management unit employed by the processor to, at least, resolve a symbolic reference in a program instruction to a memory address in the one or both of primary memory and secondary memory, during program instruction execution...", as recited in Applicants' amended claim 18. Spencer fails to say anything with respect to, for example, resolving a symbolic reference in a program instruction to a memory address

in one or both of primary memory and secondary memory, during program instruction execution, in accordance with amended claim 18.

In addition, Applicants respectfully submit that nothing in FIG. 1 and related text, nor any other portion or figure of Spencer teaches or suggest, at least "...at least one reference lookup table resident in memory accessible by the reference lookup table management unit, the at least one reference lookup table comprising one or more entries each associating a symbolic reference representing a location within a program component, to a memory address in the one or both of primary memory and secondary memory...", as recited in amended claim 18.

Applicants now turn to Spencer, at paragraph [0047], which states:

The invention can be implemented in digital electronic circuitry, or in computer hardware, firmware, software, or in combinations of them. Apparatus of the invention can be implemented in a computer program product tangibly embodied in a machine-readable storage device for execution by a programmable processor; and method steps of the invention can be performed by a programmable processor executing a program of instructions to perform functions of the invention by operating on input data and generating output. The invention can be implemented advantageously in one or more computer programs that are executable on a programmable system including at least one programmable processor coupled to receive data and instructions from, and to transmit data and instructions to, a data storage system, at least one input device, and at least one output device. Each computer program can be implemented in a high-level procedural or object-oriented programming language, or in assembly or machine language if desired; and in any case, the language can be a compiled or interpreted language. Suitable processors include, by way of example, both general and special purpose microprocessors. Generally, a processor will receive instructions and data from a read-only memory and/or a random access memory. Generally, a computer will include one or more mass storage devices for storing data files; such devices include magnetic disks, such as internal hard disks and removable disks; magneto-optical disks; and optical disks. Storage devices suitable for tangibly embodying computer program instructions and data include all forms of non-volatile memory, including by way of

example semiconductor memory devices, such as EPROM, EEPROM, and flash memory devices; magnetic disks such as internal hard disks and removable disks; magneto-optical disks; and CD-ROM disks. Any of the foregoing can be supplemented by, or incorporated in, ASICs (application-specific integrated circuits).

Applicants respectfully submit that paragraph [0047] of Spencer amounts to nothing more than a laundry list of the many types of hardware and software elements that may be used to implement the invention of Spencer. However, neither paragraph [0047] of Spencer, nor any other portion or figure of Spencer teaches or suggests, at least, "...a reference lookup table management unit employed by the processor to, at least, resolve a symbolic reference in a program instruction to a memory address in the one or both of primary memory and secondary memory, during program instruction execution...", and "...at least one reference lookup table resident in memory accessible by the reference lookup table management unit, the at least one reference lookup table comprising one or more entries each associating a symbolic reference representing a location within a program component, to a memory address in the one or both of primary memory and secondary memory...", as recited in Applicants' amended claim 18. Therefore, Spencer fails to teach or suggest at least this aspect of Applicants' invention.

Based at least upon the reasons set forth above and those previously set forth with respect to the rejection of Applicants' claim 10, Applicants respectfully submit that Spencer fails to teach or suggest all of the elements of Applicants' amended claim 18, as required by M.P.E.P. §2131, that the Office has failed to establish a *prima facie* case of anticipation, and that a rejection of amended claim 18 cannot be maintained.

Therefore, Applicants believe that amended claim 18 is allowable, for at least the reasons set forth above. Applicants respectfully submit that claims 19-25 depend from claim 18 and are therefore also allowable, for at least the same reasons. Applicants therefore respectfully request the reconsideration and withdrawal of the rejection of claims 18-25 under 35 U.S.C. 102(e).

II. Sharon Does Not Anticipate Claims 10-12, 14 and 18-25.

Applicants first address the rejection of claim 10, which has been amended as shown above to more clearly describe the subject matter of the claim.

Applicants respectfully submit that Sharon fails to teach, suggest, or disclose "...retrieving an existing version of code and a corresponding updated version of the code, the existing and the updated versions of code comprising program components each comprising program instructions...", as recited in Applicant's amended claim 10. The Office action asserts that the Sharon discloses "...retrieving an existing ... and an...updated version of...code...;" (ellipsis and strikeouts in original)(Office action at page 7) Applicant notes that the Office relies on Sharon at page 1, line 13 to page 2, line 11, which states:

Turning to a specific example of computer programs, an *old program* is installed at a remote client site and is subject to be upgraded to a *new program*, where the latter includes some modifications as compared to the old program.

In order to carry out the update at the remote client site (through the network), the provider should, preferably, generate a difference result representative of the difference between the *old program* and the *new program*, and send the resulting file through the Internet to the remote client site. The client, in turn, invokes appropriate utility, which incorporates the differences in the old program, thereby generating the desired new program at the client site. The specified procedure carries the obvious advantages in that on the one hand, the provider does not need to be present at the client site and, on the other hand, only the difference result and not the entire new program is sent to the client. Assuming, for example, that a modified *Office '97* package (commercially available from Microsoft Inc. USA) should be sent to clients, since the compressed size of programs of the package occupies tens of Mega-bytes, and, further considering the relatively low throughput of the Internet and the bottleneck of the modem throughput at the client end (say an average of 33,600 bps), it is easy to understand that transmitting the entire new package through the network is practically infeasible.

(italics in original)

The Applicants respectfully submit that neither the above portion of Sharon, which was specifically identified in the Office action, nor any other portion or figure of Sharon teaches or suggests, at least "...the existing and the updated versions of code comprising program components each comprising a plurality of program instructions...", as recited by Applicants amended claim 10. Applicants respectfully submit that neither the prior Office action (dated 2/7/07) nor the instant Office action offers any explanation or reasoning regarding which teaching of Sharon corresponds to Applicants' element "program components". The cited portion of Sharon simply explains reasons why generating a difference result between old and new programs is desirable, but neither this portion, nor any other portion or figure of Sharon provides any teaching with regard to "program components" that "each compris[e] a plurality of program instructions", in accordance with Applicants' amended claim 10. Therefore, Applicants respectfully submit that Sharon fails to teach or suggest at least this aspect of Applicants' amended claim 10.

In addition, Applicants respectfully submit that Sharon does not teach, suggest, or disclose "...generating an associated reference lookup table having entries corresponding to those program components to be modified, each entry associating a symbolic reference in the updated version of code with a memory address in the mobile electronic device...", as recited in Applicants' amended claim 10. The Office action asserts that Sharon discloses "...generating an associated reference lookup table having entries corresponding to those program components to be modified...." (Office action at page 8) Applicant respectfully disagrees.

The Office initially relies on Sharon at page 5, lines 22-29, which states:

On the basis of this observation, the invention aims at generating a modified old program and a modified new program, wherein the difference in references in corresponding entries in said new and old programs as explained above, will be reflected as invariant entries in the modified old and new programs. The net effect is that the

invariant reference entries (between the modified old program and the modified new program), will not appear in the difference result, thereby reducing its size as compared to a conventional difference result obtained by using hitherto known techniques.

Applicants respectfully submit that nothing in the portion of Sharon shown above, or in the remainder of Sharon, teaches or suggests "...generating an associated reference lookup table having entries corresponding to those program components to be modified, each entry associating a symbolic reference in the updated version of code with a memory address in the mobile electronic device...", in accordance with Applicants' amended claim 10. The cited portion makes no mention of generating a table, but instead discloses generating modified old and new programs. The Office action, at page 8, asserts that:

"...The Applicant also indicates that this feature is not taught by Sharon the feature is considered taught via the cited portions above and for further emphasis, the applicant should see page 6 (specific [sic] the references) and page 11 line 29-page 14 line 24 (specifically the data tables, lookup tables)...."

(emphasis in original)

Applicants respectfully submit that Sharon, at page 6, states:

Accordingly, the invention provides for a method for generating a compact difference result between an old program and a new program; each program including reference entries that contain reference that refer to other entries in the program ; the method comprising the steps of

- (a) scanning the old program and for substantially each reference entry perform steps that include:
 - (i) replacing the reference of said entry by a distinct label mark, whereby a modified old program is generated;
- (b) scanning the new program and for substantially each reference entry perform steps that include:

- (i) replacing the reference of said entry by a distinct label mark, whereby a modified new program is generated;
- (c) generating said difference result utilizing directly or indirectly at least said modified old program and modified new program.

The invention further provides for a method for performing an update in an old program so as to generate a new program ; each program including reference entries that contain reference that refer to other entries in the program; the method comprising the steps of

- (a) receiving data that includes a compact difference result; said compact difference result was generated utilizing a modified old program and a modified new program;
- (b) scanning the old program and for substantially each reference entry perform steps that include:
 - (i) replacing the reference of said entry by a distinct label mark, whereby the modified old

(cited text ends)

Sharon, at page 11, line 29 to page 14, line 24 states:

Yet further, the invention provides for a method for generating a compact difference result between an old data table and a new data table; each data table including reference entries that contain reference that refer to other entries in the data table; the method comprising the steps of

- (a) scanning the old data table and for substantially each reference entry perform steps that include:
 - (i) replacing the reference of said entry by a distinct label mark, whereby a modified old data table is generated;
- (b) scanning the new data table and for substantially each reference entry perform steps that include:
 - (i) replacing the reference of said entry by a distinct label mark, whereby a modified new data table is generated;
- (c) generating said difference result utilizing directly or indirectly at least said modified old data table and modified new data table.

Moreover, the invention provides for a method for performing an update in an old data table so as to generate a new data table; each data table including reference entries that contain reference that refer to other entries in the data table; the method comprising the steps of:

- (a) receiving data that includes a compact difference result; said compact difference result was generated utilizing a modified old data table and a modified new data table;
- (b) scanning the old data table and for substantially each reference entry perform steps that include:
 - (i) replacing the reference of said entry by a distinct label mark, whereby the modified old data table is generated;
- (c) reconstituting the modified new data table utilizing at least said compact difference result and said modified old data table; said modified new data table is differed from said new data table at least in that substantially each reference entry in said new data table is replaced in said modified new data table by a distinct label mark;
- (d) reconstituting said new data table utilizing directly or indirectly at least said compact difference result and said modified new data table.

The invention further provides for a method for generating a compact difference result between an old data table and a new data table; each data table including reference entries that contain reference that refer to other entries in the data table; the method comprising the steps of:

- (a) generating a modified old data table utilizing at least said old data table;
- (b) generating a modified new data table utilizing at least said new data table, said modified old data table and modified new data table have at least the following characteristics:
 - (i) substantially each reference in an entry in said old data table that is different than corresponding entry in said new data table due to delete/insert modifications that form part of the transition between said old data table and new data table are reflected as invariant references in the corresponding entries in said modified old and modified new data tables;

- (c) generating said compact difference result utilizing at least said modified new data table and modified old data table.

The invention provides for a method for performing an update in an old data table so as to generate a new data table; each data table including reference entries that contain reference that refer to other entries in the data table; the method comprising the steps of:

- (a) receiving data that includes a compact difference result; said compact difference result was generated utilizing a modified old data table and a modified new data table;
- (b) generating a modified old data table utilizing at least said old data table;
- (c) reconstituting a modified new data table utilizing directly or indirectly at least said modified old data table and said compact difference result; said modified old data table and modified new data table have at least the following characteristics:
 - (i) substantially each reference in an entry in said old data table that is different than corresponding entry in said new data table due to delete/inset modifications that form part of the transition between said old data table and new data table are reflected as invariant references in the corresponding entries in said modified old and modified new data tables;
- (d) reconstituting said new data table utilizing directly or indirectly at least said compact difference result and said modified new data table.

Again, Applicants respectfully submit that nothing in the portions of Sharon shown above, or in the remainder of Sharon, teaches or suggests "...generating an associated reference lookup table having entries corresponding to those program components to be modified, each entry associating a symbolic reference in the updated version of code with a memory address in the mobile electronic device...", in accordance with Applicants' amended claim 10. The cited portions make no mention of generating a "reference lookup table", or of "program components", in accordance with Applicants' amended claim 10. The above text of Sharon does mention an "old data table", a "modified old data table", a "new data table", and a "modified new data table"

that are used as inputs to “generating said compact difference result”. The Office action states that “...**the applicant should see page 6 (specific [sic] the references) and page 11 line 29-page 14 line 24 (specifically the data tables, lookup tables).**” (page 8) Applicants respectfully submit, however, that Sharon makes no mention of a table used for reference lookup (“lookup tables” as suggested in the Office action), and that the “data tables” generated by Sharon (i.e., “modified old data table” and “modified new data table”) are different from and do not teach or suggest Applicants’ “reference lookup table having entries corresponding to those program components to be modified, each entry associating a symbolic reference in the updated version of code with a memory address in the mobile electronic device”, as recited in Applicants’ amended claim 10. Therefore, Applicants respectfully submit that Sharon fails to teach or suggest at least this aspect of Applicants’ amended claim 10.

Based at least upon the above, Applicants respectfully submit that the Sharon fails to teach or suggest each and every element of Applicants’ amended claim 10, as required by M.P.E.P. §2131, that the Office has failed to establish a *prima facie* case of anticipation, and that a rejection of amended claim 10 under 35 U.S.C. §102(b) cannot be maintained.

Therefore, Applicants believe that amended claim 10 is allowable over Sharon, for at least the reasons set forth above. Applicants respectfully submit that claims 11-14 depend from claim 10, and are also allowable, for at least the same reasons. Applicants therefore respectfully request that the rejection of claims 10-14 under 35 U.S.C. §102(b) be reconsidered and withdrawn.

Applicants next address the rejection of claim 18, which has been amended to more clearly describe the subject matter of the claim. The instant Office action rejects claim 18 “...as claim 1, in view of page 1 lines 5-15 and claims [sic] 16 in view of claim 21.” (Office action at page 9) Again, Applicants respectfully submit that there is no pending claim 1. Applicants also respectfully request that the Office explain the intended meaning of the portion of the rejection which states that claim 18 is rejected in view of “...claims [sic] 16 in view of claim 21.”

The Office further action states "...[t]he electronic device is considered to consist of the processor and the primary and secondary memory; since items like the lookup table and the management unit does [sic] not create or enable a new device. Furthermore software stored in memory also does not enable a new device and the processor and memories appear to be individual components that do not interact until claim 20. **The applicant indicates that Sharon does not teach the management unit; however, this is the feature that is considered to enable step (c) in lines 24-29 of page 15. The lookup tables are listed above.**" (emphasis in original)(Office action at page 9) Applicants respectfully disagree with the statements cited above. Applicants respectfully submit that the rejection of claim 18 has been rendered moot by Applicants' amendments.

Applicants respectfully submit that Sharon fails to teach, suggest, or disclose, at least, "...a reference lookup table management unit employed by the processor to, at least, resolve a symbolic reference in a program instruction to a memory address in the one or both of primary memory and secondary memory, during program instruction execution...", as recited in Applicants' amended claim 18. Applicants respectfully submit that Sharon is silent with respect to resolving a symbolic reference in a program instruction to a memory address, during execution of the program instruction, in accordance with Applicants' amended claim 18. Instead, Sharon relates to "...a method for generating a compact difference result between an old program and a new program...." (Abstract) Applicants respectfully disagree with the statement in the Office action that "...items like the lookup table and the management unit does [sic] not create or enable a new device." (Office action at page 9) Applicants respectfully submit that the Office has failed to set forth any basis in 35 U.S.C., 37 C.F.R., or the M.P.E.P. for the apparent dismissal of limitations of Applicants' claim 18. Based at least upon the above, Applicants respectfully submit that Sharon fails to teach or suggest at least these aspects of Applicants' amended claim 18.

In addition, Applicants respectfully submit that Sharon fails to teach, suggest, or disclose "...at least one reference lookup table resident in memory accessible by the reference lookup table management unit, the at least one reference lookup table

comprising one or more entries each associating a symbolic reference representing a location within a program component, to a memory address in the one or both of primary memory and secondary memory...”, as recited in Applicants’ amended claim 18. Sharon is simply silent with respect to a “lookup table” having the recited content and used in this manner, in accordance with amended claim 18. And as discussed above, Sharon fails to teach or suggest a “program component”, let alone “a symbolic reference representing a location within a program component”, in accordance with Applicants’ amended claim 18.

Based at least upon the reasons set forth above, and also those previously set forth with respect to the rejection of Applicants’ claim 10, Applicants respectfully submit that Sharon fails to teach or suggest all of the elements of Applicants’ amended claim 18, as required by M.P.E.P. §2131, that the Office has failed to establish a *prima facie* case of anticipation, and that a rejection of amended claim 18 cannot be maintained.

Therefore, Applicants believe that amended claim 18 is allowable, for at least the reasons set forth above. Applicants respectfully submit that claims 19-25 depend from claim 18 and are therefore also allowable, for at least the same reasons. Applicants therefore respectfully request the reconsideration and withdrawal of the rejection of claims 18-25 under 35 U.S.C. 102(b).

III. Combining Either Spencer Or Sharon with Yang Does Not Render Claims 13 and 26-32 Unpatentable

Applicants respectfully submit that Yang is not a valid reference, in accordance with 35 U.S.C. 103(c), and that the Office has failed to establish a *prima facie* case of obviousness, as required by 35 U.S.C. §103(a).

Applicants respectfully submit that, in accordance with 35 U.S.C. 103(c), Yang is not a valid reference in the rejection of claims 13 and 26-32 under 35 U.S.C. 103(a), because the present application and Yang (U.S. Patent Application Ser No. 10/635,991 which published as U.S. Patent Application Publication No. 2004/0040020 A1) were, at the time the invention was made, owned by, or under a

common obligation to assign ownership to, Bitfone Corporation, now a wholly-owned subsidiary of Hewlett-Packard Development Corporation, LLC.

Applicants set forth this same argument in the previous response. (Amendment filed May 22, 2007, pages 15-16) In response to that traversal, the instant Office action states “...[t]he applicant claims that Yang and the present invention was [sic] commonly owned; however, it is not clear that the inventions were co-owned at the time of the inventions. That is, no affidavits have been filed in the present application and nothing in the applicant’s [sic] statements indicates when Bitfone Corporation became a wholly owned subsidiary of Hewlett Packard Corp. Therefore, the previous rejection remains.” (Office action at page 6) Applicants respectfully submit that the Applicants did not file any affidavits or any statements providing evidence of common ownership or obligation to assign, because the M.P.E.P. expressly states that Applicants are not required to submit such affidavits, or other evidence of common ownership or obligation to assign.

Applicants respectfully submit that M.P.E.P. §706.02(I)(2)(II) states, in part:

The following statement is sufficient evidence to establish common ownership of, or an obligation for assignment to, the same person(s) or organizations(s):

Applications and references (whether patents, patent applications, patent application publications, etc.) will be considered by the examiner to be owned by, or subject to an obligation of assignment to the same person, at the time the invention was made, if the applicant(s) or an attorney or agent of record makes a statement to the effect that the application and the reference were, at the time the invention was made, owned by, or subject to an obligation of assignment to, the same person.

See “Guidelines Setting Forth a Modified Policy Concerning the Evidence of Common Ownership, or an Obligation of Assignment to the Same Person, as Required by 35 U.S.C. 103(c),” 1241 O.G. 96 (December 26, 2000). The applicant(s) or the representative(s) of record have the best knowledge of the ownership of their application(s) and reference(s), and their statement of such is sufficient evidence because of their paramount obligation of candor and good faith to the USPTO.

The statement concerning common ownership should be clear and conspicuous (e.g., on a separate piece of paper or in a separately labeled section) in order to ensure that the examiner quickly notices the statement. **Applicants may, but are not required to, submit further evidence, such as assignment records, affidavits or declarations by the common owner, or court decisions, in addition to the above-mentioned statement concerning common ownership.**
(emphasis added)

Applicants respectfully submit that the statement of common ownership set forth in the response filed May 22, 2007, repeated above, is both clear and conspicuous, and accurately states that the present application and Yang were, at the time the invention was made, owned by, or under a common obligation to assign ownership to, Bitfone Corporation. Applicants respectfully submit that the Office has failed to set forth any material evidence that has been indicated to raise a doubt as to the accuracy of Applicants' representation of either (1) the common ownership of, or (2) the existence of an obligation to commonly assign, the application being examined and the applied U.S. patent or U.S. patent application publication reference. The mere disclosure, by the Applicants, that Bitfone Corporation is now a wholly-owned subsidiary of Hewlett-Packard Development Corporation, LLC, does not constitute material evidence of a lack of candor and good faith.

Applicants respectfully submit that the acquisition of Bitfone Corporation by Hewlett-Packard Company was completed on February 7, 2007, and that this information is publicly available at:

<<http://h71028.www7.hp.com/enterprise/cache/524119-0-0-225-121.html>>

as of October 31, 2007. Applicants respectfully submit that Yang was filed August 7, 2003, that the instant application was filed September 3, 2003, and that the completion of the acquisition of Bitfone Corporation by Hewlett-Packard Company was completed more than three years after the filing of the instant application and Yang.

Therefore, Applicants respectfully submit that the Applicants have met the requirements of M.P.E.P. §706.02(I)(2)(II), that Yang is not a valid reference in accordance with 35 U.S.C. §103(c), that the Office has failed to establish a *prima facie* case of obviousness as required by 35 U.S.C. §103(a), and respectfully request that the rejections of claims 13, and 26-32 under 35 U.S.C. 103(a) in view of Spencer and Yang, and Sharon and Yang, be reconsidered and withdrawn.

Conclusion

In general, the Office Action makes various statements regarding the claims and the cited references that are now moot in light of the above. Thus, Applicants will not address such statements at the present time. However, Applicants expressly reserve the right to challenge such statements in the future should the need arise (e.g., if such statements should become relevant by appearing in a rejection of any current or future claim).

The Applicants believe that all of claims 10-14 and 18-32 are in condition for allowance, and courteously solicit a Notice of Allowability.

Should the Examiner disagree or have any questions regarding this submission, the Applicants invite the Examiner to telephone the undersigned at (312) 775-8000.

Respectfully submitted,

Dated: November 21, 2007

Hewlett-Packard Company
Intellectual Property Administration
Legal Department, M/S 35
P.O. Box 272400
Fort Collins, CO 80527-2400

/Kevin E. Borg/

Kevin E. Borg
Reg. No. 51,486